IEEE International Conference on Systems, Man & Cybernetics (SMCC), Washington D C, USA, 2003: 3816-3821

Source code: http://www.adaptivebox.net/research/fields/algorithm/pso/index.html

DEPSO: Hybrid Particle Swarm with Differential Evolution Operator

Wen-Jun Zhang, Xiao-Feng Xie*
Institute of Microelectronics,
Tsinghua University,
Beijing 100084, P. R. China
*Email: xiexiaofeng@tsinghua.org.cn

Abstract - *A hybrid particle swarm with differential evolution operator, termed DEPSO, which provide the bell-shaped mutations with consensus on the population diversity along with the evolution, while keeps the self-organized particle swarm dynamics, is proposed. Then it is applied to a set of benchmark functions, and the experimental results illustrate its efficiency.*

Keywords: Particle swarm optimization, differential evolution, numerical optimization.

1 Introduction

Particle swarm optimization (PSO) is a novel multi-agent optimization system (MAOS) inspired by social behavior metaphor [12]. Each agent, call *particle*, flies in a D-dimensional space S according to the historical experiences of its own and its colleagues. The velocity and location for the ith particle is represented as $\vec{v}_i$ = ($v_{i1},\ldots,v_{id},\ldots,v_{iD}$) and $\vec{x}_i = (x_{i1},..., x_{id},..., x_{iD})$, respectively. Its best previous position is recorded and represented as $\vec{p}_i$=(p_{i1}, …, p_{id}, …, p_{iD}), which is also called *pbest*. The index of the best *pbest* is represented by the symbol g, and $\vec{p}_g$ is called *gbest*. At each step, the particles are manipulated according to the following equations [15]:

$$v_{id} = w\cdot v_{id}+c_1\cdot rand()\cdot(p_{id}-x_{id})+c_2\cdot rand()\cdot(p_{gd}-x_{id}) \quad (1a)$$

$$x_{id} = x_{id} + v_{id} \quad (1b)$$

where w is inertia weight, c_1 and c_2 are acceleration constants, *rand*() are random values between 0 and 1.

Several researchers have analyzed it empirically [1, 11, 20] and theoretically [3, 5], which have shown that the particles oscillate in different sinusoidal waves and converging quickly, sometimes prematurely, especially for PSO with small w[20] or constriction coefficient[3].

The concept of a more-or-less permanent social topology is fundamental to PSO [10, 12], which means the *pbest* and *gbest* should not be too closed to make some particles *inactively* [8, 19, 20] in certain stage of evolution. The analysis can be restricted to a single dimension without loss of generality. From equations (1), v_{id} can become small value, but if the $|p_{id}-x_{id}|$ and $|p_{gd}-x_{id}|$ are both small, it cannot back to large value and lost exploration capability in some generations. Such case can be occured even at the early stage for the particle to be the *gbest*, which the $|p_{id}-x_{id}|$ and $|p_{gd}-x_{id}|$ are zero, and v_{id} will be damped quickly with the ratio w. Of course, the lost of diversity for $|p_{id}-p_{gd}|$ is typically occured in the latter stage of evolution process.

To maintain the diversity, the DPSO version [20] introduces random mutations on the x_{id} of particles with small probability c_l, which is hard to be determined along with the evolution, at least not be too large to avoid disorganization of the swarm. It can be improved by a bell-shaped mutation, such as Gaussian distribution [8], but a function of consensus on the step-size along with the search process is preferable [11]. A bare bones version [11] for satisfying such requirements is to replace the equations (1) by a Gaussian mutation with the mean $(p_{id}+p_{gd})/2$ and the standard deviation $|p_{id}-p_{gd}|$, which maybe also be inefficient when $|p_{id}-p_{gd}|$ is very small, and is too radically since it turns the PSO into a variaty of in evolution strategies (ES) [2].

This paper describes a hybrid particle swarm with differential evolution (DE) operator[16], termed DEPSO, which also provide the bell-shaped mutations with consensus on the population diversity, while keeps the particle swarm dynamics. Then the DEPSO is applied to several benchmark functions [4, 13, 15], and the results illustrate the significant performance improvement.

2 DEPSO algorithm

For proposed DEPSO, the mutations are provided by DE operator [16] on the $\vec{p}_i$, with a trail point $\vec{T}_i = \vec{p}_i$, which for dth dimension:

IF ($rand()<CR$ **OR** $d==k$) **THEN** $T_{id} = p_{gd} + d_{2,d}$ (2)

where k is a random integer value within [1, D], which ensures the mutation at least one dimension., CR is a crossover constant, and $\vec{d}_2$ is the case of N=2 for the general difference vector $\vec{d}_N$, which is defined as:

$$\vec{d}_N = \left(\sum_1^N \vec{\Delta}\right) \Big/ N \qquad (3)$$

where the $\vec{\Delta}$ is the essential difference vector [16], means the difference of two elements that random chosen from a common point set, which include all the *pbest* in current case. N is the number of $\vec{\Delta}$ involved. Then for the dth dimension:

$$\vec{\Delta}_d = \vec{p}_{A,d} - \vec{p}_{B,d} \qquad (4)$$

where $\vec{p}_A$, $\vec{p}_B$ are chosen from the *pbest* set at random.

The experimental analysis f or $\vec{d}_N$ will be restricted to a single dimension without loss of generality. Fig. 1 and 2 shows a histogram of points that were tested in one million iterations for $\vec{d}_1$ and $\vec{d}_2$, respectively. Each one dimensional element $\vec{p}$ for calculating $\vec{\Delta}$ is a real-value that picked from [-1, 1] at random. It can be seen that $\vec{d}_1$ is a triangle distribution and $\vec{d}_2$ is a bell-shaped distribution, which the latter is better for problem-solving [11]. Hence the N=2 is selected in equation (2).

The mutation is performed on the *pbest* instead of $\vec{x}_i$ [8, 20] so as to prevent the swarm from disorganizing by unexpected fluctuations, since the replacement of *pbest* will follow the steady-state strategy, i.e., $\vec{T}_i$ will replace $\vec{p}_i$ only if it is better than $\vec{p}_i$.

The mutation is also based on $\vec{p}_g$ provides the social learning capability, which might speed up convergence. The learning ratio is determined by CR, which is the counterpart of interaction probability IP [11]. If CR=1, then the equation (2) becomes a bell-shaped mutation operator on $\vec{p}_g$ as ES [2]. If CR<1, it may retain some dimensions of $\vec{p}_i$, which will facilitate the convergence for the fitness landscape that some dimensions are irrelevantly. Such capability is also implemented implicitly in the some former mutation versions [8, 20] but is not in the canonical version.

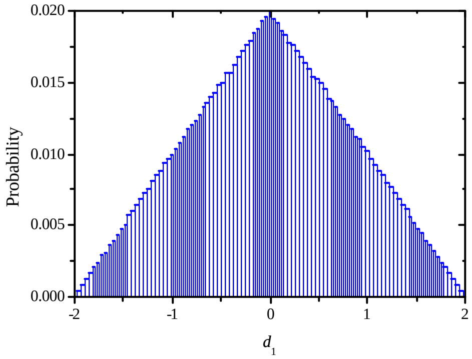

FIG. 1 Histogram of points tested for $\vec{d}_1$

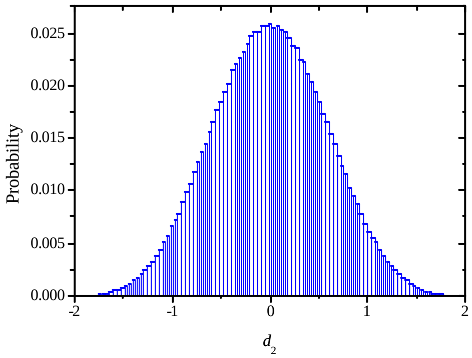

FIG. 2 Histogram of points tested for $\vec{d}_2$

The original PSO operator and the DE operator will be performed alternately, i.e. the equations (1) will be performed at the odd generations, and the equation (2) at the even generations. The $\vec{d}_2$ will provide a consensus mutation on $\vec{p}_i$ along with diversity of swarm, which emerge from the nature of the search itself, while trying to keep the diversity of *pbest* and *gbest* by changing $\vec{p}_i$.

3 Results and discussions

For DEPSO, c_1=c_2=2, w=0.4, the maximum velocity V_{MAX} was set to the half range of the search space on each dimension [9, 20]. If without declaration specially, each test case ran for T=2000 generations and all the cases were run 100 runs each.

The constraint-handling method is following the criteria [6]: a) any feasible solution is preferred to any infeasible solution; b) among two feasible solutions, the one having better objective function value is preferred; c) among two infeasible solutions, the one having smaller constraint violation is preferred.

The boundary constraints are handled by *Periodic* mode. For each point $\vec{x}_i$, its fitness will be calculated by a *mapping point* $\vec{z}_i = (z_{i1},..., z_{id},..., z_{iD})$, which for dth dimension, it has:

$$\textbf{IF } x_{id}< l_d \textbf{ THEN } z_{id}= u_d -(l_d -x_{id})\% s_d \qquad (5a)$$

$$\textbf{IF } x_{id}> u_d \textbf{ THEN } z_{id}= l_d +(x_{id}- u_d)\% s_d \qquad (5b)$$

Where '%' is the modulus operator, l_d and u_d are lower and upper values, and s_d =| u_d - l_d | is the parameter range of the dth dimension.

3.1 Unconstrainedfunctions

The tested unconstrained problems include the Rosenbrock function, the generalized Rastrigrin and the generalized Griewank function [15, 19, 20], which all in 30-D and have same minimumvalue (F_{opt}) as zero.

Table 1 lists the mean fitness values of such functions by FPSO [15], DPSO (w=0.4, c_l=0.001) [20], and DEPSO (w=0.4, CR=0) with different number of agents (m). The CR is set small, since there is little correlation between the parameters for these functions. It shows that DEPSO performs better than both old PSO versions, especially for the Rastrigrin and the Griewank functions, which with uncorrelated parameters.

TABLE 1. The mean fitness values for the unconstrained functions

f	m	FPSO[15]	DPSO[20]	DEPSO
30-D Rosenbrock (F_{opt}=0)	20	183.8037	132.1512	80.8259
	40	175.0093	82.7209	66.8730
	80	124.4184	57.2802	60.6405
30-D Rastrigrin (F_{opt}=0)	20	48.47555	7.3258	0.8656
	40	35.20146	6.2107	0.009950
	80	22.52393	4.2265	3.919E-9
30-D Griewank (F_{opt}=0)	20	0.021560	0.01793	0.009073
	40	0.012198	0.01356	0.006930
	80	0.014945	0.01190	0.005589

3.2 Constrained functions

The selected problems include eleven functions that are proposed by Z. Michalewicz et al (G_1 to G_{11}) [13], which include eight functions without equality constraints and three functions (i.e. G_3, G_5, G_{11}) with equality constraints, and an engineering optimization problem: design of a pressure vessel (*Vessel*) [4, 9]. The dimension of S (D), the optimization type and optimum value (F_{opt}) of each function are list in Table 2.

TABLE 2. Summary of constrained functions

f	D	Type	F_{opt}
G_1	13	Minimum	-15
G_2	20	Maximum	0.803612
G_4	5	Minimum	-30665.539
G_6	2	Minimum	-6961.814
G_7	10	Minimum	24.306
G_8	2	Maximum	0.095825
G_9	7	Minimum	680.630
G_{10}	8	Minimum	7049.248
G_3	10	Minimum	-1
G_5	4	Minimum	5126.498
G_{11}	2	Minimum	0.75
Vessel	4	Minimum	6059.714

Table 3 lists the mean fitness values for eight functions without equality constraints by a (30, 200)-ES (T=1750) [17], the DE (CR=0.1), the canonical PSO (w=0.4), and the DEPSO (w=0.4, CR=0.1), where m=70. For G_8, the maximum generation T=200, and for all the other cases, T=2000. It shows that the DEPSO outperforms either the DE or the PSO. By the way, it also provides better results than GA [13] and ES [17] with much less evaluation times.

TABLE 3. The F_{opt} for the functions without equality constraints

f	ES[17]	DE	PSO	DEPSO
G_1	-15.000	-15.000	-14.945	-15.000
G_2	0.7820	0.7805	0.6891	0.7868
G_4	-30665.5	-30650.1	-30665.5	-30665.5
G_6	-6875.94	-6961.81	-6961.81	-6961.81
G_7	24.374	25.064	25.286	24.586
G_8	0.095825	0.095825	0.095825	0.095825
G_9	680.656	681.063	680.652	680.641
G_{10}	7559.192	7565.4	7526.6	7267.4

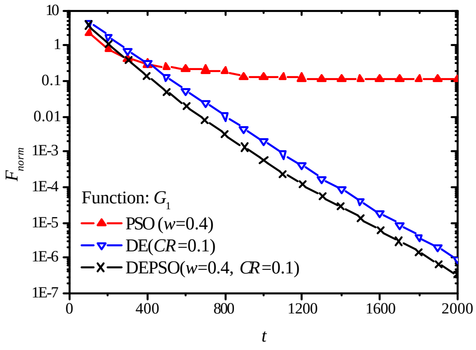

FIG. 3 Mean relative performance for G_1

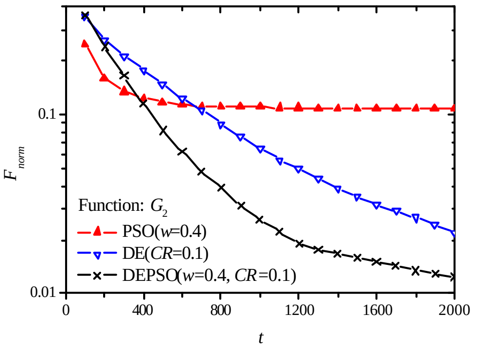

FIG. 4 Mean relative performance for G_2

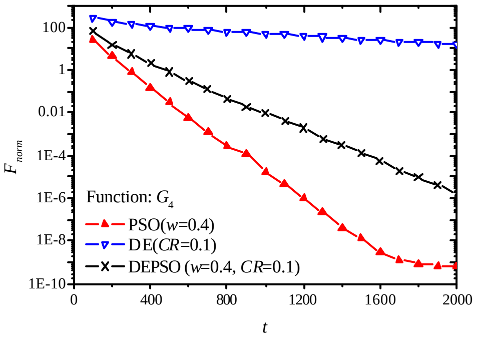

FIG. 5 Mean relative performance for G_4

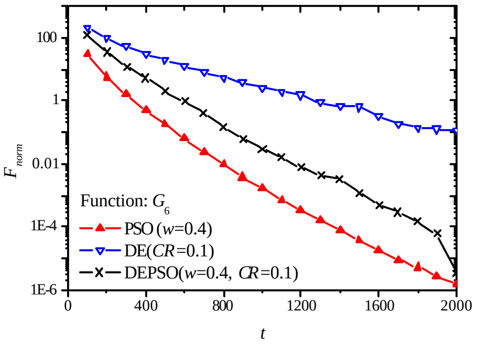

FIG. 6 Mean relative performance for G_6

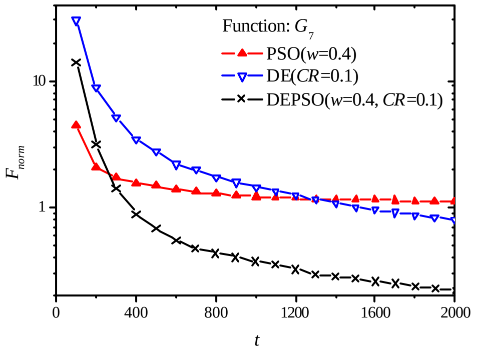

FIG. 7 Mean relative performance for G_7

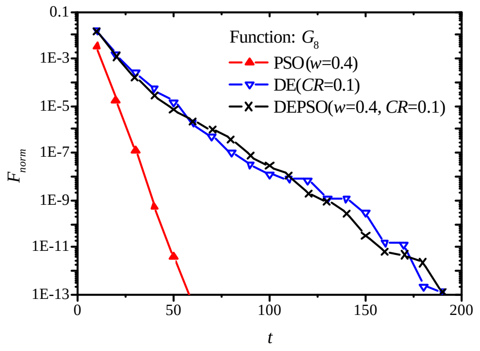

FIG. 8 Mean relative performance for G_8

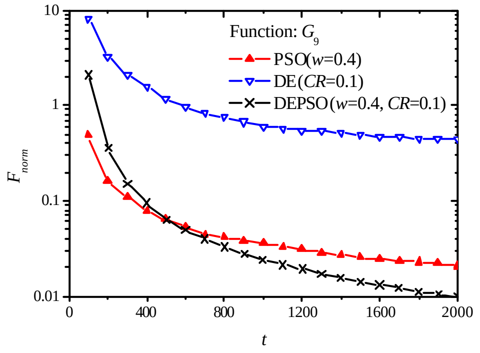

FIG. 9 Mean relative performance for G_9

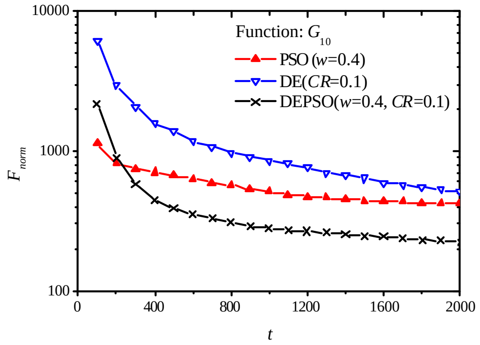

FIG. 10 Mean relative performance for G_{10}

Figure 3 to 10 show the relative mean fitness value F_{norm}=F_{mean}-F_{opt} during the evolution generations (t) for the eight functions, respectively, where F_{mean} is the mean best fitness value in each generation. It shows that DEPSO outperform DE in at least mentioned cases. For some problems, such as G_4, G_6, G_8, PSO perform better than DE and DEPSO with CR=0.1, which may due to the fitness landscape of such problems are *epistasis*, i.e. need to covary the parameters at the same time to improve fitness. Notes although there is no correlation between the parameters of G_6, but its constraints construct such a S_F. This also can explain why PSO is performing worse for such as G_1, since lack of the capability of varying only few dimensions for a point.

For G_3, G_5, G_{11}, which have almost 0% feasible space (S_F) due to the equality constraints, are needed to replace the constraint $g(\vec{x})=0$ by an inequality constraint $|g(\vec{x})|<e$ for some small $e>0$ [13, 14, 17]. Here we choose two e values: 1E-3 [14], and 1E-4 [17], which the S_F of the former is larger than that of the latter and will more easily to be solved.

Table 4 lists the mean fitness values for the three functions with equality constraints DE (CR=0.9), the canonical PSO (w=0.4), and DEPSO (w=0.4, CR=0.9), where m=70. In order to compare with existing results, when e =1E-3, for G_5, T=2500, and for G_{11}, T=300; when e =1E-4, for G_3 and G_5, T=5000. For other cases, T=2000. The values in the brackets gives the number of trails that are failed in entering S_F, and only those trails that are succeeded in entering S_F are counted for the calculation of mean fitness values. DEPSO outperforms DE and PSO in all cases, and it can find the optimum solution in all runs when e =1E-3. Table 5 summarizes the evaluation times for existing DE results in [14] and DEPSO, which shows DEPSO is also much fast.

TABLE 4. The F_{opt} for the functions with equality constraints

f	e	DE	PSO	DEPSO
G_3	1E-3	-0.8572	-1.0048	-1.0050
G_3	1E-4	-0.3985	-0.8364	-0.9849
G_5	1E-3	5129.50	5356.15(10)	5126.484
G_5	1E-4	5133.834	5334.97(12)	5130.864
G_{11}	1E-3	0.74909	0.74941	0.74900
G_{11}	1E-4	0.75061	0.75459	0.74990

TABLE 5. Summary of the evaluation times when e =1E-3

f	DE[14]	DEPSO
G_3	8000000	140000
G_5	1200000	175000
G_{11}	30000	21000

When e =1E-4, DEPSO also cannot always find the optimum point, which is better than normal ES [17] but is worse than ES with stochastic ranking [17] for G_3 and G_5 in same evaluation times.

The objective of the *vessel* problem is to minimize the cost of the material, forming and welding of a cylindrical vessel [4, 9]. It is a mixed-integer-discrete-continuous problem which has four design variables, two are integer and two are continuous. Here the closest integer value will be used to evaluate the fitness although the algorithms still works internally with continuous variables.

TABLE 6. The mean fitness values for the *vessel* problem

Results	MGA[4]	DE	PSO	DEPSO
Best	6069.3267	6062.740	6059.714	6059.714
Mean	6263.7925	6126.658	6332.784	6108.177
Worst	6403.4500	6408.792	6820.410	6410.087
S.D.	97.9445	76.0619	263.4612	93.45319

Table 6 lists the comparison of results for the *vessel* problem by MGA (population size is 50, and T=1000, which costs 50000 evaluation times) [4], DE (CR=0.1), the canonical PSO (w=0.4), and DEPSO (w=0.4, CR=0.1), where for the three algorithms, m=70, and T=700, which costs 49000 evaluation times. It can be seen that the PSO is not stable [9], which may due to the step-type landscape created by integer variables. However, PSO also shows the capability to catch the optimum point comparing to MGA and DE. DEPSO inherits the merits of PSO and DE, and performs much better that MGA within almost same evaluation times.

4 Conclusions

This paper presents a hybrid particle swarm with differential evolution operator called DEPSO. The hybrid strategy provides the bell-shaped mutations with consensus on the population diversity by DE operator, while keeps the self-organized particle swarm dynamics, in order to make the performance is not very sensitive to the choice of the strategy parameters as in DE [7]. It is shown to outperform the PSO and DE for a set of benchmark functions. However, more comparative works with different parameter settings for more problems should be performed to provide a full view.

The DEPSO seems performing well for the problems with integer variables by the help of the bell-shaped mutations. However, as declared for PSO [9], it is also not very efficiently for handling those problems with extremely small feasible space, such as the problems with equality constraints. Since each agent in DEPSO (and DE, PSO) only refers to few points (*pbest* and *gbest*), it cannot employs some strategies (such as stochastic ranking [17]), which needs a big population. Future investigation may employ extending memory with a set of points to satisfy such strategies.

Moreover, according to No-Free-Lunch (NFL) theory [18], taken the problem information into account will improve the performance of algorithm. For DEPSO, the appropriate *CR* can be chosen if the correlation of the parameters is known. But for the black-box problems, it is still a great challenge to learning such parameters during run-time with efficiently strategies.

Acknowledgements

The work is supported by the National Hi-Tech Research and Development Program of China (863 project) under Contract No. 2002AA1Z1460. The authors also thank M. Clerc for helpful discussions.

References

[1] Angeline P J. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference. Annual Conf. on Evolutionary Programming, 1998: 601-610

[2] Beyer H-G, Schwefel H-P. Evolution strategies: a comprehensive introduction. Natural Computing, 2002, 1(1): 3-52

[3] Clerc M, Kennedy J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. on Evolutionary Computation, 2002, 6(1): 58-73

[4] Coello C A C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering, 2002, 191(11-12): 1245-1287

[5] Cristian T I. The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters, 2003, 85(6): 317–325

[6] Deb K. An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2-4): 311-338

[7] Gäemperle R, Mü ller S D, Koumoutsakos P. A parameter study for differential evolution. WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, 2002: 293-298

[8] Higashi N, Iba H. Particle swarm optimization with Gaussian mutation. IEEE Swarm Intelligence Symposium, 2003: 72-79

[9] Hu X H, Eberhart R C, Shi Y H. Engineering optimization with particle swarm. IEEE Swarm Intelligence Symposium, 2003: 53-57

[10] Kennedy J. The particle swarm: social adaptation of knowledge. IEEE Int. Conf. on Evolutionary Computation, 1997: 303-308

[11] Kennedy J. Bare bones particle swarms. IEEE Swarm Intelligence Symposium, 2003: 80-87

[12] Kennedy J, Eberhart R C. Particle swarm optimization. IEEE Int. Conf. on Neural Networks, 1995: 1942-1948

[13] Koziel S, Michalewicz Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evolutionary Computation, 1999, 7(1): 19-44

[14] Lampinen J. A constraint handling approach for the differential evolution algorithm. Congress on Evolutionary Computation, 2002: 1468-1473

[15] Shi Y H, Eberhart R C. Fuzzy adaptive particle swarm optimization. IEEE Int. Conf. on Evolutionary Computation, 2001: 101-106

[16] Storn R, Price K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optimization, 1997, 11: 341–359

[17] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization. IEEE Trans. on Evolutionary Computation, 2000, 4(3): 284-294

[18] Wolpert D H, Macready W G. No free lunch theorems for optimization. IEEE Trans. on Evolutionary Computation, 1997, 1(1): 67-82

[19] Xie X F, Zhang W J, Yang Z L. Adaptive particle swarm optimization on individual level. Int. Conf. on Signal Processing, 2002: 1215-1218

[20] Xie X F, Zhang W J, Yang Z L. A dissipative particle swarm optimization. Congress on Evolutionary Computation, 2002: 1456-1461